REMARKS

Applicants respectfully request reconsideration of the present application based on the foregoing amendments and the following remarks. Applicants herein amend claims 30 and 31. Claims 1-12 and 14-33 remain pending in the application.

Claim Rejections Under 35 U.S.C. § 102(a)

Claims 1-2 and 14-29 stand rejected under 35 U.S.C. 102(a) as being anticipated by Jain et al., "ASIP Design Methodologies: Survey and Issues" ("Jain"). For reasons set forth more fully below, this rejection is respectfully traversed.

Independent Claim 1 Patentably Defines Over Jain

Independent claim 1 requires, *inter alia*:

> means for **automatically generating an instruction set architecture** optimized for executing that program(s), wherein the instruction set architecture is represented as a set of configurations **containing one or more extension instructions based on instructions in** an existing standard or **existing user defined instruction set architecture**.

This subject matter is not taught or suggested by Jain, at least because Jain does not disclose or suggest anything about an **existing user defined instruction set architecture**, much less generating an instruction set architecture containing one or more extension instructions based on such **existing user defined instruction set architecture**.

As set forth in the present specification beginning at, for example, page 9, line 19, such an existing **user-defined** instruction set architecture can comprise an Xtensa ISA that has been extended with user-defined operations using the Tensilica Instruction Extension (TIE) language as described in U.S. patents and applications commonly owned by the present assignee. The specification further explains that, in this example embodiment, the invention creates new operations by applying vector operation, fused operation, and specialization techniques to such existing operations, and then uses VLIW to create new instructions composed of one or more original or new operations.

Jain merely provides a survey of ASIP design methodologies, which include methodologies for selecting architectures and instruction sets for a particular application. At

pages 79-80, Jain teaches that instruction sets can be either synthesized from an analysis of operations used in an application, or selected from a superset of instructions. Nowhere does Jain disclose or suggest anything about creating extension instructions **based on existing instructions** in any type of instruction set architecture, much less "**automatically generating an instruction set architecture** . . . **containing one or more extension instructions based on instructions in** an . . . **existing user defined instruction set architecture**" as expressly required by independent claim 1.

For at least this reason, claim 1 patentably defines over Jain, and the § 102 rejection of claim 1, as well as claims 2 and 14-29 that depend from claim 1, should be withdrawn.

Dependent Claims 2 and 14-29 Contain Additional Limitations Not Taught By Jain

As set forth above, claims 2 and 14-29 depend directly or indirectly from claim 1 and thus patentably define over Jain for at least the same reasons as claim 1. Furthermore, claims 2 and 14-29 contain additional subject matter not taught or suggested by Jain.

For example:

- Claim 2 requires that extension instructions "operate on states . . . in the existing user-defined instruction set architecture." The Jain reference has been thoroughly reviewed, and no disclosure of instructions operating on states was located in the reference at all, much less operating on user-defined states as required by claim 2.
- Claim 16 requires analysis information including execution counts "determined from user-supplied directives." The specification teaches at, for example, page 20, lines 3-5 that a user-supplied directive is inserted into the application by using a pragma, for example. The Jain reference has been thoroughly reviewed, and no disclosure of user-supplied directives was located in the reference at all.
- Claim 17 requires that the analysis information includes "a dependence graph of each region." The closest possible language identified in the Jain reference was the consumption of a scalar "in the data flow graph" on page 77. This does not disclose or suggest analyzing an application by dependence graph of regions of code as required by claim 17.

- Claim 18 requires that the analysis information includes "a set of operation vector lengths that can be used to improve performance of each region." The Jain reference has been thoroughly reviewed, and no disclosure of using a set of operation vector lengths to improve performance of a region of code was located in the reference at all. The closest possible match was using "loop vectorization" as an optimization technique in page 78.
- Claims 23-26 require providing performance improvement determined by "an instruction scheduling algorithm operating on a modified dependence graph of the region." The closest possible language identified in the Jain reference was the consumption of a scalar "in the data flow graph" on page 77. This does not disclose or suggest modifying a dependence graph of regions of code as required by claims 23-26.

For at least these reasons, dependent claims 2 and 14-29 further patentably define over Jain.

Claim Rejections Under 35 U.S.C. § 102(a)

Claims 3-12 and 30-33 stand rejected under 35 U.S.C. 103(a) as being obvious in view of Jain and the prior art of record. For reasons set forth more fully below, this rejection is respectfully traversed.

Claims 3-12 depend directly or indirectly from claim 1, and thus are patentable for at least the same reasons claim 1 is patentable. Independent claims 31-33 also require operating upon a **existing user-defined instruction set architecture** and thus are patentable at least for similar reasons as claim 1.

Claims 3-12 and 30-33 further variously require **automatically generating extension instructions or register files** for an **existing standard or user-defined instruction set architecture** that are either vectorized, VLIW combinations, fused combinations and/or specialized **versions of existing instructions or register files in the architecture**.

The Office Action admits that this subject matter is not taught by Jain. However, the Office Action takes the position that since vector operations, VLIW instructions, fused

operations, and specialized instructions were known at the time of the invention, this subject matter would have been obvious.

This position is incorrect, because Jain does not disclose or suggest **automatically generating extension instructions** for an **existing instruction set architecture** at all, and no prior art exists for operating using an existing **user-defined** instruction set architecture. Because Jain does not suggest this subject matter required by the claims, the alleged combination of Jain with the instruction techniques of the specification prior art would not have suggested this subject matter either.

For at least these reasons, claims 3-12 and 30-33 patentably define over the cited prior art and the § 103 rejections of these claims should be withdrawn.

Conclusion

All objections and rejections having been addressed, it is believed that the present application is in condition for allowance, and a Notice to that effect is earnestly solicited. If any issues remain which the Examiner feels may be resolved through a telephone interview, s/he is kindly requested to contact the undersigned at the telephone number listed below.

Respectfully submitted,
PILLSBURY WINTHROP LLP

Date: November 5, 2004

40,580
Reg. No.

Mark J. Danielson
(650) 233-4777
Please reply to customer no. 27,498